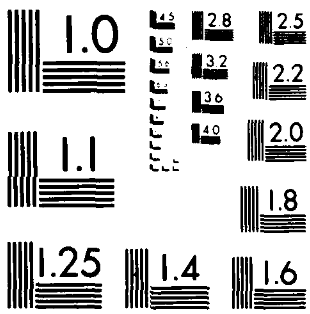
AD-A135 154 SEARCH ALGORITHMS AND THEIR IMPLEMENTATION(U) DUKE UNIV DURHAM NC DEPT OF COMPUTER SCIENCE D W LOVELAND AUG 83 AFOSR-TR-83-0932 AFOSR-81-0221 1/1

UNCLASSIFIED F/G 12/1 NL

END
DATE
FILMED
12 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

A135154

(4)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83-0932	2. GOVT ACCESSION NO. A135154	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SEARCH ALGORITHMS AND THEIR IMPLEMENTATION		5. TYPE OF REPORT & PERIOD COVERED Annual Report 6/30/82 - 6/29/83
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) D.W. Loveland		8. CONTRACT OR GRANT NUMBER(s) AFOSR 81-0221
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department Duke University Durham, NC 27706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM BOLLING AFB, DC 20332		12. REPORT DATE August 1983
		13. NUMBER OF PAGES 15
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Annual Report

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Research that has resulted in completed papers includes (1) perfect information games of chance, (2) non-minimax search strategies, (3) limited resource search and (4) knowledge evaluation in expert systems. Preliminary results have been obtained in areas including (a) parallel search algorithms, (b) test-and-treatment problems and (c) search using expectation. Other investigations are in progress.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

Research that has resulted in completed papers includes (1) perfect information games of chance, (2) non-minimax search strategies, (3) limited resource search and (4) knowledge evaluation in expert systems. Preliminary results have been obtained in areas including (a) parallel search algorithms, (b) test-and-treatment problems and (3) search using expectation. Other investigations are in progress.

ANNUAL REPORT

Search Algorithms and Their Implementation
(AFOSR 81-0221)

June 30, 1982 - June 29, 1983
(Second Year)

Overview

This is a summary of the second year of the Air Force Office of Scientific Research grant 81-0221. It has been a productive year with most projects continuing research begun last year and an effort initiated at the beginning of this year in parallel search algorithms. Newer yet, but underway, are explicit studies in the search heuristics associated with parsing error-prone input and general ambiguity resolution. Several reports on different aspects of our research have been written, with most intended for publication. Papers summarizing specific research results are to be given at the *International Joint Conference on Artificial Intelligence* in Karlsruhe, West Germany, and at the *Third National Conference on Artificial Intelligence* in Washington, both in August, 1983. In addition, several conference presentations have already occurred this year.

The grant supports research on several facets of search problems, including abstract models of search and applications of search. Summaries of the separate efforts are included as individual sections. We are pleased at the joint collaboration between the specific study of test-and-treatment search algorithms, in particular computationally fast approximation algorithms, and the study in parallel search algorithms for the Boolean Vector Machine, a parallel architecture machine being developed at Duke. The test-and-treatment problem is a very general fault analysis problem, with readily-derivable branch-and-bound and dynamic programming solutions, but these algorithms are computationally very time-consuming. The parallel search group is seeking a relatively fast parallel implementation of the dynamic pro-

gramming solution of this fault analysis problem to complement the search for fast (sequential) approximation algorithms.

The personnel supported by the AFOSR grant remains quite stable, the graduate students included. Most of the flux has been in the parallel algorithms endeavor for various reasons. Andrew Reibman, who is shifting from work on non-minimax strategies to parallel algorithm study, is the major change. We used much more computer resources (primarily for simulation studies) than planned, in contrast to last year, and expect this to be true this coming year also. In general, the support provides the positive environment that allows productive research, and we are optimistic regarding the year now underway.

Research Objectives

Our research objective remains a multifaceted study of search techniques and applications. The research on non-minimax search strategies and parallel algorithms for the test-and-treatment problem may be winding down, the general study of test-and-treatment problems, the limited resource search study and the knowledge evaluation research in expert systems continues, new work in search techniques in parsing is underway, and investigation of new parallel search algorithms is to begin. This reflects the successful conclusion of certain projects, the new opportunities discovered, and the continuing challenge of other projects.

Research Status

The status report is divided into projects with the personnel involved named in parentheses. Graduate students' names appear before their advisor's name when the advisor's role is limited to problem formulation and research guidance.

Perfect information games of chance (Ballard, Reibman)

Further studies of the *-minimax search algorithms for trees containing chance nodes were conducted, and two hybrid algorithms derived from those presented at AAAI-82 were developed and tested. A paper was accepted for publication in *Artificial Intelligence* and incorporates the results of these most recent studies as well as our previous ones.

Non-mimimax search strategies (Ballard, Reibman)

We continued our studies of our *-Min search procedure, for which preliminary results were given in the most recent Air Force proposal. Papers describing these studies were presented at the *21st Southeast Region ACM Conference* and at the Army-sponsored *Conference on Artificial Intelligence* in Rochester, Michigan.

A new study of the role of opponent error was then begun by Reibman, and a paper giving a model of player fallibility, and preliminary results on its expected performance, presented at the *Third National Conf. on Artificial Intelligence* in August, 1983.

We also began to study the more elaborate "D-PBU" search strategy we had formulated during the spring of 1982, which was motivated by the "product rule" methods proposed by Pearl which have subsequently been studied by him and Nau. Collecting together all the ideas and strategies proposed by ourselves, as well as recent proposals of Nau and Pearl and early work by Slagle and Dixon, Ballard undertook a comprehensive series of empirical studies of several non-minimax strategies. The principal results of these studies are: (1) all non-minimax criteria studied yield a slight but significant improvement over minimax; (2) strategies based upon a simple average do better than those based on products; (3) the most likely, and in some cases the most important, opportunity to outperform minimax arises when minimax is faced with a tie; (4) departures from minimax tend to occur in unfavorable positions; and (5) non-minimax strategies exhibit uneven performance when pitted against one another. These results, along with a definition of the non-minimax strategies that were

considered and an account of the historical relation among them, has very recently been written. It became a Technical Report in late July and will be submitted for publication in August, 1983.

Further work on the efficacy of *-Min as a function of player error has been done by Reibman and will be submitted for journal publication in August or September.

Maximizing payoff from limited resource search (Mutchler, Loveland)

The first year of work on this problem led to a number of insights concerning the case of extremely limited resources for search (i.e., only one "fuzzy snapshot" of the environment was allowed). This second year our efforts on this single view case was completed and a lengthy report written on this case. The next research step has been underway for roughly half of the year. This step, allowing two "fuzzy snapshots" of the environment is considerably more difficult to handle analytically so a computer program for simulations has been written. The program makes it relatively easy to test various search and move strategies.

We outline the structure of this second step. The general task is to find good search and move strategies for finding a path down a complete (say, binary) tree with values at each leaf. We seek to find the highest value or as high a value as is possible. For a unit expenditure (say, one dollar) one can name a node and be told the summary distribution of values in the subtree with that node as root; however, no location information of values is returned. One is given a limited number of dollars and so certainly wishes to employ a good strategy for the search process. After the search is completed, he may move to whatever node he wishes (which may be the root node of the entire tree if the search information is discouraging; the player is always given the distribution at the root). From then on all moves are random; at present we take each path in the subtree as equally likely. (One exception: when moving the player may specify to delete from consideration any subtree headed by a searched node.) Again,

one seeks a good strategy for selecting search nodes. Other variations of the above description are interesting and to be considered. These tasks are all simplifications of limited resource tasks in general. For more motivation see the report "Search with limited resources" by D. Mutchler.

The step now being explored is the two-dollar resource search, i.e., two queries for subtree distribution. Not surprisingly, the simulations show a much more complex strategy situation than for the one-dollar search problem. While no pattern for an optimal strategy is yet apparent, we have a tentative working conjecture that one good strategy is to spend the first dollar at a node just before a leaf. If an acceptably high value is in the subtree use the second dollar to locate it by quizzing one of the two leaves. Else try outside this subtree in like manner. This is a good strategy for the "single treasure" (one non-zero value) case.

Over this next year we hope to develop a good understanding of good (ideally, optimal) strategies for multidollar search, and, with luck, a proof of this, if necessary under constrained conditions.

Binary testing: tests and treatment (Loveland)

The object of binary testing is to find optimal search trees, called decision trees, to find one distinguished object among many objects. In the first year of the grant (a year ago) we were successful in extending some results of Garey and Graham from a limited case (equiprobable *a priori* probabilities on the candidate objects) to the general case (arbitrary *a priori* probabilities). While studying this problem we realized that this was not the right problem for fault diagnosis because one does not always test to isolation. Rather, as one may well be able to treat the problem at an intermediate stage, by repairing (replacing) a collection of objects which includes the faulty object such that the cost of this repair is cheaper than isolation of the primitive component and its repair. This is a very interesting and pertinent problem for which no

work exists (at least that is known to experts in the binary testing area). A major task is to formulate the right model so that solutions will be interesting. We have formulated a general model which already makes some assumptions, such as that the tests and treatments are replicable and deterministic e.g., a treatment either cures the problem or not (*versus* a probabilistic statement pertaining to how a specific treatment interacts with a specific object). Professor Wagner and a student have formulated the general dynamic programming representation of this general model to study parallel computation algorithms finding optimal test-and-treatment trees when tests and treatments have arbitrary cost. This is to be implemented on the Boolean Vector Machine. (See the entry for Professor Wagner.) Because this problem includes the binary testing problem we know it is NP-hard in its general formulation. However, when the set of tests and treatments is sufficiently rich, a polynomial-time algorithm may exist. In the binary test subcase with all test costs the same it is well known that there is such an algorithm -- the Huffman Code Algorithm, which is quite fast ($n \log n$ where n is the number of candidates). We are seeking the analogous result with all test costs equal and treatment costs proportional to the size of the set they treat. Several first-step results have been obtained, mostly relating to limiting values for the proportionality constant. The first non-trivial interesting result has just been obtained (actually this month, on the current-year grant): we have showed that for this treatment cost pattern an optimal treatment tree must have tests always preceding treatments, regardless of the *a priori* probability distribution *or* the proportionality constant. (Note that a treatment need not be a leaf because if the distinguished object is not in the treatment set then one must continue processing, by tests and/or treatments.)

Discussions with Bruce Ballard have led to a somewhat more general model for our analytic work. Time is needed to see if it is amenable to analytic results or if it is too rich to tackle, at least until more knowledge is obtained for simpler models first.

Parallel search algorithms (Wagner, Duval, Han)

During the past several months, significant progress has been made in designing efficient parallel algorithms for the Test and Treatment Problem, which is also under investigation by Loveland. Our effort has been concentrated on the development of algorithms for the Boolean Vector Machine for this problem. The machine in question is designed as an array of many thousands of processing elements, connected by a general permutation network called the Cube Connected Cycles network, a well-known network algorithm. It uses bit-serial algorithms and communication links. Nonetheless, it is expected to outperform a conventional machine of equal cost by roughly the ratio of the number of words in the conventional machine's memory to 8 times the width of those words.

The performance advantage of the BVM can only be achieved by redesign of algorithms, to take advantage of massive parallelism at the algorithm level. We are attempting this for the computationally intensive dynamic programming algorithm which solves the Test and Treatment problem exactly. We hope that the speed-up achieved is sufficient that experimental comparison of the "true" answers to examples of this problem can be made with the heuristic answers of Loveland's methods. In addition, there is some hope of being able to use the exact algorithm for modest-size problems without spending years awaiting the answers.

To date, we have designed one parallel algorithm for this problem. Dan Duval has written a "C" program which generates instructions for the BVM, and is proceeding to test the resulting BVM programs, by using a functional simulator of the machine. Debugging of this algorithm is proceeding, as is the writing of a paper describing its development, and comparing its projected performance on the BVM with the performance of a logically-similar program for a conventional machine. Dan recently left the project to work at MCNC, and has been replaced by Yijie Han, who is scheduled to complete Dan's program, and assist Dan and Robert Wagner in writing a paper describ-

ing it. We currently believe that the algorithm will execute in time O(n**2 * log n * p) on a problem with n objects, and weights expressed as integers of p bits in length.

Recently, another algorithm for this problem has been devised by Wagner, running in time O(n * log n * p), for a machine whose interconnection network is a Boolean Hypercube. The CCC network allows the simulation of certain algorithms for an equal-size network of Processing Elements connected by the Boolean Hypercube, at a loss in time of a factor of 4. Unfortunately, the algorithm in question does not seem to fall in this simulatable class. Yijie Han is scheduled to investigate whether, in fact, this type of algorithm can be made to run on the BVM at the same speed as it does on the hypercube (to within a small constant factor).

Perhaps just as important as speed is the question of the number of PE's needed for a problem of given size. This number corresponds roughly to the number of words of memory needed on a conventional machine. The algorithm investigated by Dan Duval requires n**2*2**n PE's as written, and so becomes infeasible on a 2**16 PE array for very small values of n. An algorithm running on n*2**n PE's might well be preferable. In addition, these algorithms should be generalized, to take advantage of a smaller than optimal PE array, at a corresponding sacrifice in time. These investigations will be begun, but probably not completed, during the coming few months.

Knowledge evaluation for expert systems (Loveland, Valtorta)

Knowledge evaluation is a part of knowledge acquisition for expert systems. Here we focus on integrating incoming (new) knowledge with the existing knowledge base. To test by standard means often means searching a large set of inputs to determine that no bad side effects have occurred by the introduction of the new piece of information, as well as to check that the intended good effect is realized. We seek methods to reduce the amount of search necessary to determine that the added information does integrate well with the existing knowledge base. Our present attention is on

rule-based systems.

Our summary of last year is still quite valid in this area of research. We reported that we had results in a particular case of interest -- classification systems -- and that some unanswered questions remained to be resolved before results could be summarized in a paper. This initial exploration was successfully completed, with the result a paper "Deleting ambiguities: an example in knowledge evaluation" (to be presented at IJCAI '83) and a new algorithm for finding "critical sets" used to support the ambiguity check. We think that the algorithm for finding critical sets has independent interest and we intend to submit it for publication.

Our example, although having some sophistication in design, is quite a limited example. In particular, there is no allowance for inexact reasoning and, in present form, requires a monotonicity condition that may be violated. One immediate goal is to relax these limitations. A general goal is to continue to develop the methodology.

We conclude this section by summarizing the example we developed and commenting on the methodology we seek. As in program verification, we seek to confirm that certain properties are maintained as the new knowledge is added. The property we have considered first is classification uniqueness for meaningful atomic input. For example, if someone has only headache symptoms he should be diagnosed as having only a headache, no matter what other new information is added. Of course, if he also has broken toe symptoms one must expect a multiple diagnosis of headache and broken toe. Therefore, we can only ask to preserve uniqueness of classification for minimal meaningful input, the "atomic" meaningful inputs, because superposition of inference is to be expected as illustrated above. The problem: how to detect minimal meaningful input vectors. This clearly is impossible in any precise sense because meaning is captured by the entire rule base, if at all. What we do is to find *minimal* ambiguous input vectors all of whose components are in two or more minimal input vectors for different classifications. Such vectors are displayed to the user; these will

be few in number but a superset of any atomic meaningful but ambiguous vectors. Finding the minimal vectors is the role of the Critical Set algorithm, which is as fast as could be expected, providing that there is not a certain kind of "tangle" of minimal sets. This example procedure provides a fast testing procedure where otherwise a great number of inputs might have to be checked.

In general, we would like a uniform methodology where one writes specifications for a property and then this property is shown to hold in some sense. This is a theorem-proving endeavor, whereas our result just described uses much less machinery, actually only a "backflooding" capability through the inference tree and minor changes in input format. We were surprised that this important property yielded to such a fast processing algorithm (low degree polynomial in the number of rules *versus* exponential growth rate for general theorem proving techniques). Perhaps specialized techniques can handle whole classes of properties and cumbersome theorem-proving techniques can be avoided in many cases.

Search in the presence of expectation (Biermann, Fink)

A common view of a search process envisions the exhaustive exploration of level after level of sprouting branches without guidance except for local pruning where possible. We examine the possibility of an overviewing mechanism which contains probabilistic information about the likelihoods of certain kinds of success. The mechanism influences the search to prefer certain paths which are more likely to be successful using historical information.

We are exploring this kind of search, for example, in the parsing of sentences spoken to a machine but recognized by an error-prone voice processor. The approach assumes a user is carrying out routine repetitive tasks so that the same or similar commands appear again and again in the dialog. The early occurrences of such statements must be clearly spoken so that errors are minimized. But as the dialog contin-

ues, the historical record of interactions enables the system to predict utterances. This enables the user to speak much more quickly and to still obtain the desired behavior despite increased error rates by the voice recognizer.

Supported Personnel

Loveland, Donald W.	(Principal Investigator)
Ballard, Bruce W.	(Co-principal Investigator)
Biermann, Alan W.	(Co-principal Investigator)
Wagner, Robert A.	(Co-principal Investigator)
Duval, D.	(Research Assistant, part-time)
Han, Yijie	(Research Assistant, part-time)
Mutcher, David	(Research Assistant)
Reibman, Andrew	(Research Assistant)
Valtorta, Marco	(Research Assistant)

Publications and Reports

July, 1981 - June, 1983

Chronologically ordered

1. Biermann, A., J. Fairfield and T. Beres. Signature tables and learning. *IEEE Trans. on Systems, Man and Cybernetics* (to appear).

2. Biermann, A. Dealing with search. *Automatic Program Construction Techniques* (Eds. Biermann, Guiho, Kodratoff). MacMillan Publ. Co. (to appear).

3. Ballard, B. A search procedure for perfect information games of chance. *Artif. Intelligence* (to appear).

4. Loveland, D. Performance bounds for binary testing with arbitrary weights. Submitted to *Acta Informatica*.

5. Loveland, D. Finding critical sets. Duke C.S. Report CS-1982-23, Nov., 1982. To be submitted for publication.

6. Mutchler, D. Search with limited resources. Duke C.S. Report CS-1983-1, Jan., 1983.

7. Ballard, B. Non-minimax search strategies for minimax trees: theoretical foundations and empirical studies. Duke C.S. report CS-1983-13, July, 1983. (submitted for publication)

Conference Presentations

July 1981 - August 1983

1) Ballard, B. A search procedure for perfect information games of chance. *Second National Conf. on Artif. Intell. - 82*, Pittsburgh, Aug. 1982.

2) Loveland, D. Knowledge acquisition and evaluation. Army Conf. on AI Application to Battlefield Info. Systems, Silver Springs, Md., April, 1983.

3) Ballard, B.W. and A.L. Reibman. What's wrong with minimax? *1983 Conf. on Artif. Intell.*, Rochester, Mich., April, 1983.

4) Reibman, A.L. and B.W. Ballard. Competition against fallible opponents. *21st ACM Southeast Region Conf.*, Durham, N.C., April, 1983.

5) Reibman, A.L. and B.W. Ballard. Non-minimax search strategies for use against fallible opponents. *Third Natl. Conf. on Artif. Intell.*, Washington, D.C., August, 1983.

6) Loveland, D. Detecting ambiguity: an example of knowledge evaluation. *Eighth Intern. Joint Conf. on Artif. Intell.*, Karlsruhe, W. Germany, August, 1983.

7) Valtorta, M. A result on the computational complexity of heuristics for the A^* algorithm. *Eighth Intern. Joint Conf. on Artif. Intell.*, Karlsruhe, W. Germany, August, 1983.